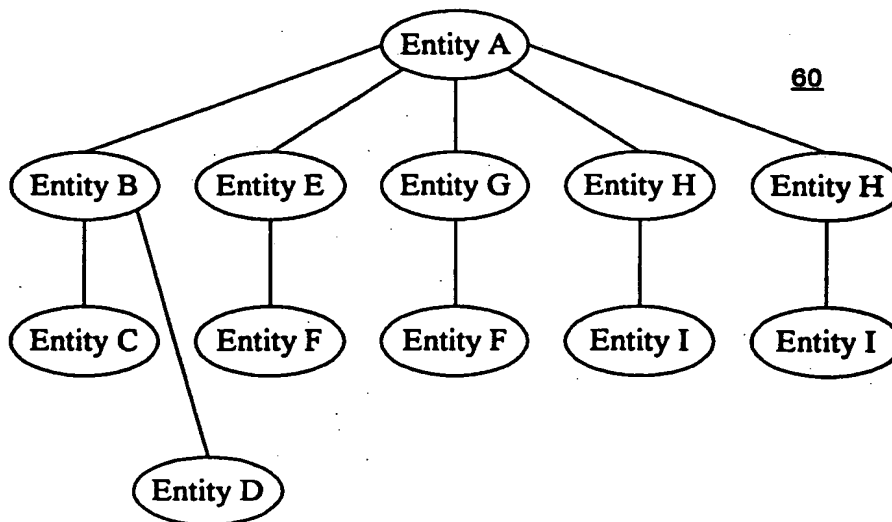


Fig. 1

54'

```
<Entity A>
  <Entity B>
    <Entity C> Data </Entity C>
    <Entity D> Data </Entity D>
  </Entity B>
  <Entity E>
    <Entity F> Data </Entity F>
  </Entity E>
  <Entity G>
    <Entity F> Data </Entity F>
  </Entity G>
  <Entity H>
    <Entity I> Data </Entity I>
  </Entity H>
  <Entity H>
    <Entity I> Data </Entity I>
  </Entity H>
</Entity A>
```

Fig. 2A

60

Fig. 2B

62

A-id	B-id	C-id	C-data	D-id	D-data	E-id	G-id

64

E-id	F-id	F-data

52'

66

G-id	F-id	F-data

68

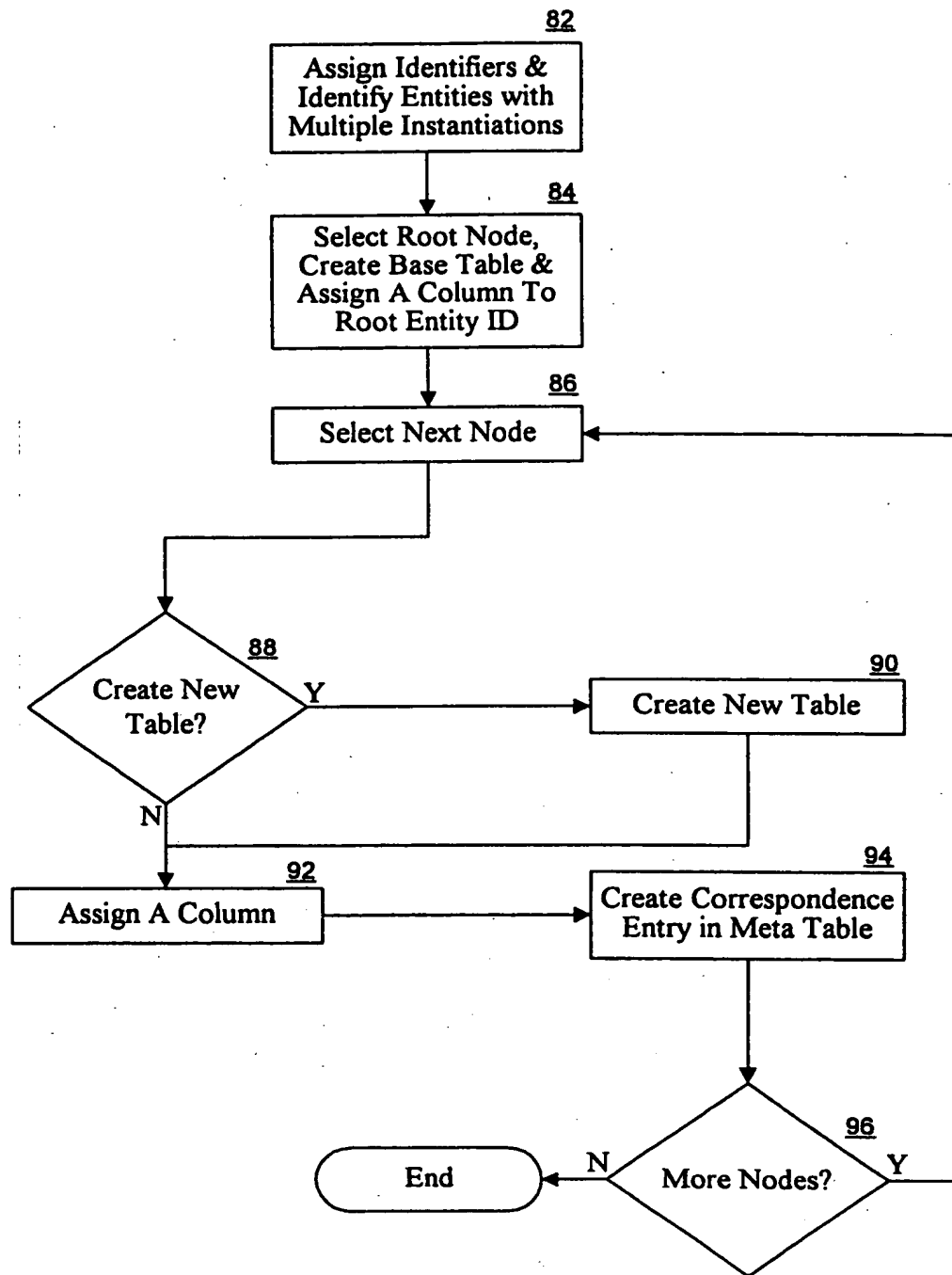
A-id	H-id	I-id	I-data

Fig. 3

56'

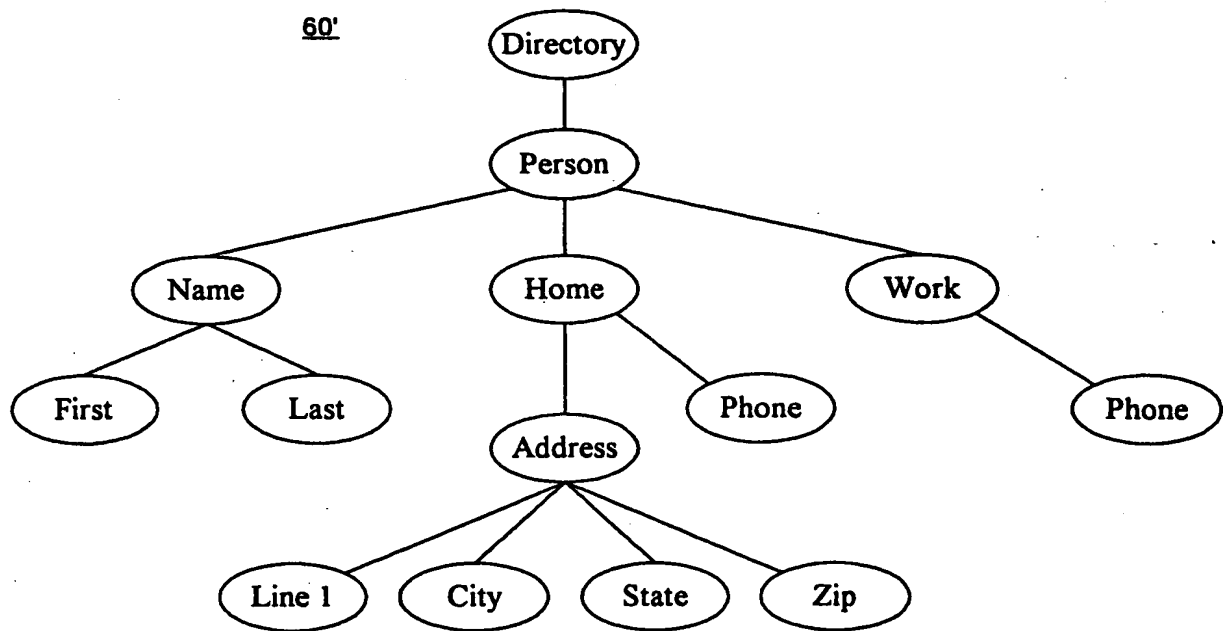
In-Context 74a	Parent Node 74b	Child Node 74c	Out-Context 74d		Table 74e	Parent Column 74f	Child Column 74g
← 72 →							

Fig. 4

*Fig. 5*

```
<directory> (100)
  <person> (102)
    <name> (104)
      <first> John </first> (107, 124)
      <last> Public </last> (108, 126)
    </name> (128)
    <home> (105)
      <address> (110)
        <line 1> 123 Main St </line1> (112,130)
        <city> Anyville </city>(114, 132)
        <state> WA </state> (116,134)
        <zip> 98198 </zip>(118,136)
      </address> (138)
      <phone> 206-255-1234 </phone> (120, 140)
    </home> (142)
    <work> (106)
      <phone> 206-555-578 </phone> (122, 144)
    </work> (146)
  </person> (148)
</directory> (150)
```

Fig. 6A

***Fig. 6B***

190

Dir-id	Per-id	Nm-id	Fn-id	Fn-data	...	Home-id	...	Work-id

192

Home-id	Ph-id	Ph-data

52'

194

Work-id	Ph-id	Ph-data

Fig. 7

56"

In-Context 74a	Parent Node 74b	Child Node 74c	Out-Context 74d		Table 74e	Parent Column 74f	Child Column 74g
	person	name			dir-tbl	per-id	nm-id
	name	first			dir-tbl	nm-id	fn-id
	first	data			dir-tbl	fn-id	fn-data
	"	"	"		"	"	"
	home	phone	h1		hm-tbl	hm-id	ph-id
h1	phone	data			hp-tbl	ph-id	ph-data
	work	phone	W1		wp-tbl	wk-id	ph-id
w1	phone	data			wp-tbl	ph-id	ph-data

Fig. 8

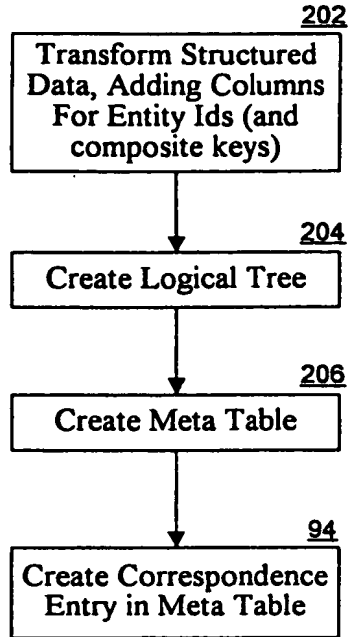


Fig. 9

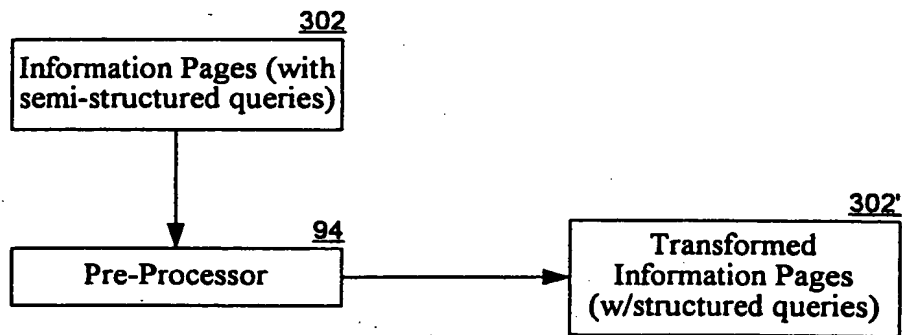
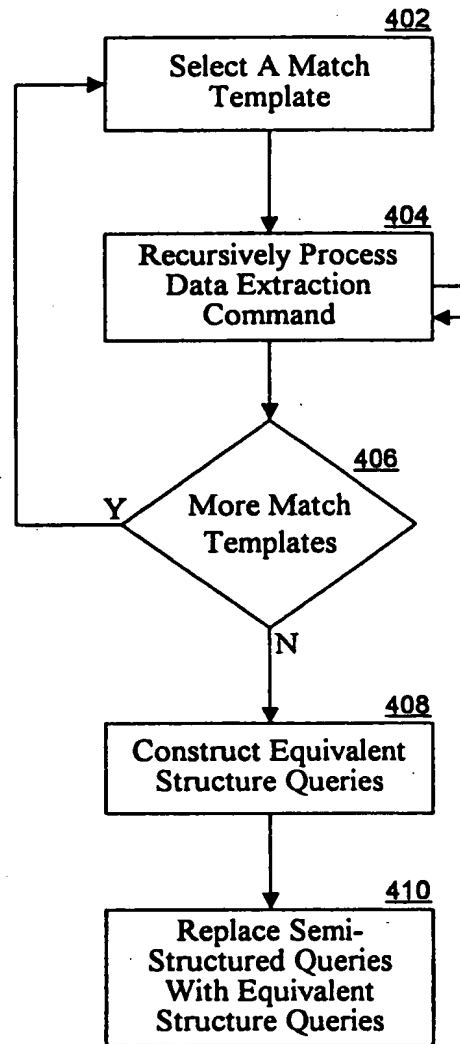


Fig. 10

***Fig. 11***

```

<xsl:stylesheet xmlns:xsl=http://www.w3.org/XSL/Transform/1.0>
  <xsl:template match="/">
    <result>
      <xsl: apply-templates
        select="companyrecords/employees/employee"/>
    </result>
  </xsl:template>

  <xsl:template match="employee">
    <xsl:variable name="depid" select=department/@idref/>
    <person>
      <xsl: apply-templates select="name"/>
      <number><xsl:value-of select="phone"/></number>
      <depname>
        <xsl: apply-templates
          select="//department[@id=$depid]/name"/>
      </depname>
    </person>
  </xsl:template>

  <xsl:template match="employee/name">
    <xsl:value-of select="first" />
    <xsl: text> </xsl:text>
    <xsl:value-of select="last" />
  </xsl:template>

  <xsl:template match="department/name">
    <xsl:value-of select="." />
  </xsl:template>

</xsl:stylesheet>

```

Fig. 12A

504

```

<companyrecords>
  <employees>
    <employee>
      <name><first>fname</first><last>lname</last></name>
      <ssn>ssn</ssn>
      <department idref=deptid/>
      <phone>xxx-xxxx</phone>
    </employee>
  </employees>
  <departments>
    <department id=deptid>
      <name> dept name </name>
    </department>
  </departments>
</companyrecords>

```

Fig. 12B506

```

(QueryLoop "Select"      name.first, name.last, employee.phone, name2.pdata,
  From                  companyrecords, multiemployee,
                        Employee, name, companyrecords as cr2,
  Where                 companyrecords.employees = multiemployee.id AND
                        cr2.departments = multidepartment.id AND
                        multidepartment.department = department.id AND
                        department.id = employee.idref,
                        department.name = name2.id)

(CreateElement "result")
  (Group "") // empty group operation groups all inside 'result'
    (CreateElement "person")
      (CreateText"SQL.1 $QL.2")
      (CreateElement "phone" body="$QL.3")
      (CreateElement "depname" body="$QL.4")

```

Fig. 12C

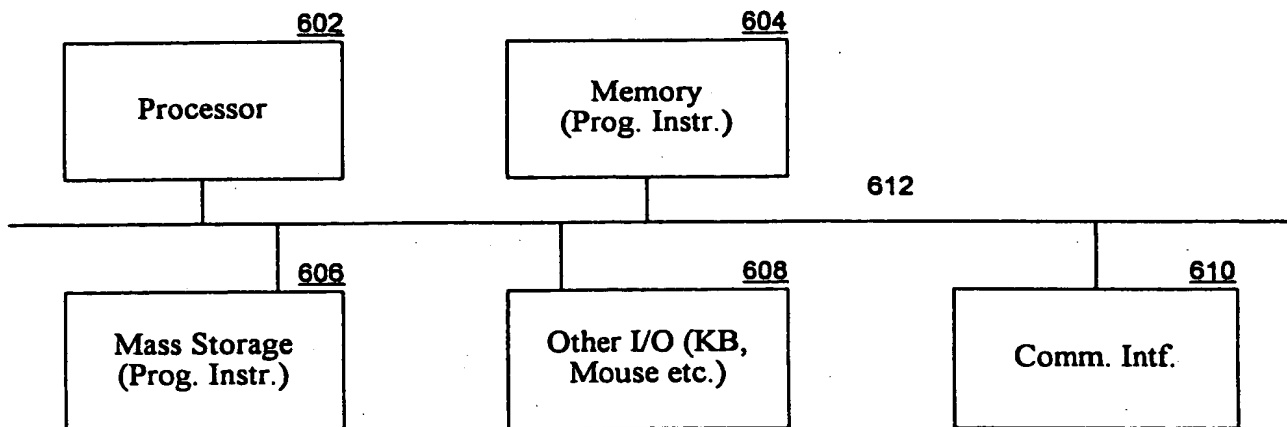


Fig. 13

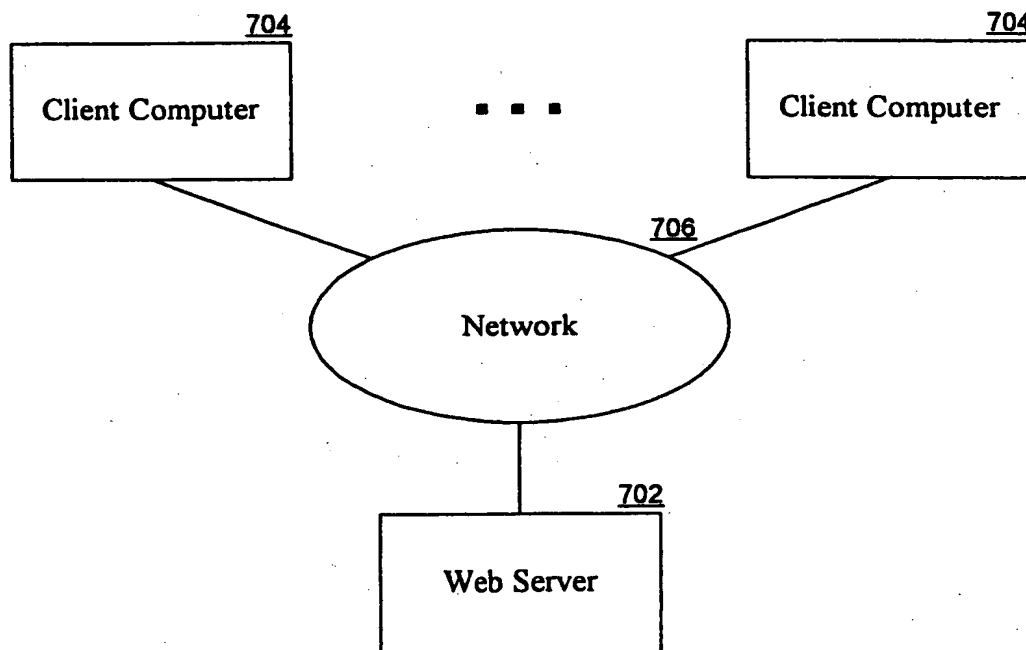


Fig. 14

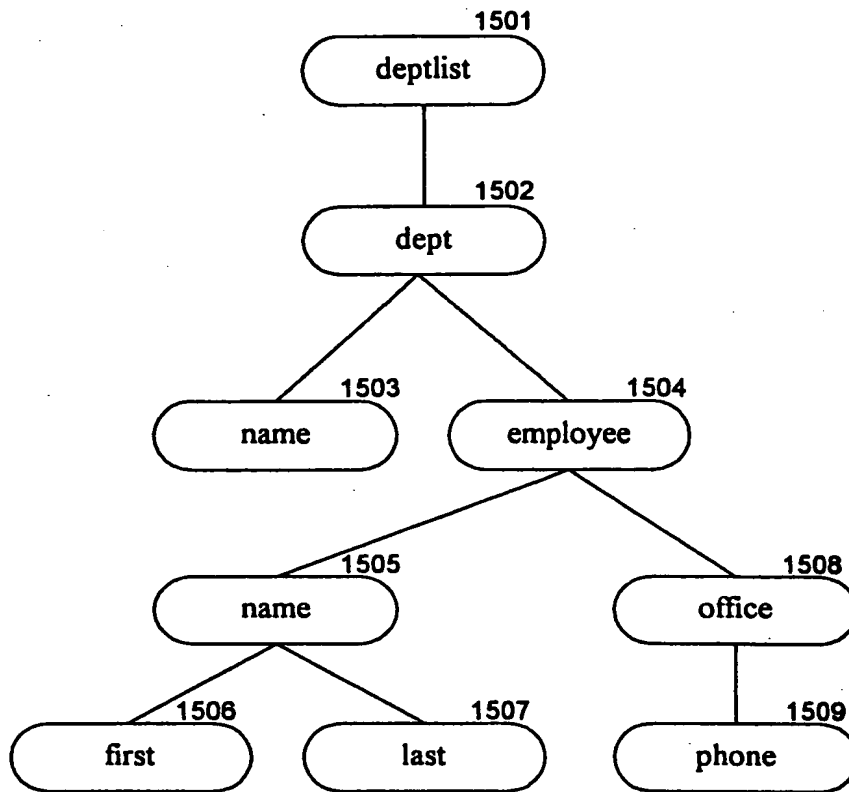


Fig. 15